# Web-oriented Data Formats and Their Management in the Mobile Era

Serena Pastore

Astronomical Observatory of Padova, National Institute of Astrophysics (INAF)
vicolo Osservatorio 5, 35122 – Padova, Italy

*1serena.pastore@oapd.inaf.it

***Abstract***

Future Internet envisages a global network of Internet-enabled objects, in most cases mobile and wireless nodes, mainly accessible through web standards. This network connection implies a big flux of data through mobile devices and wireless networks, most of which are unstructured. Such data need to be structured, collected and/or organized in web-oriented formats to be published within every web framework and managed by a backend database structure. Focusing on the handling of digital objects and mobile devices to publish information, this paper discusses how to structure and then manage the data derived from different sources in a web-oriented way. The project's starting point is to implement a device able to capture, collect, and transmit different kinds of information related to a target object. The research framework is the applications of future Internet technologies for culture dissemination and outreach. Constraints such as technologies to structure the data, ways to enrich the data with semantic information using metadata, and storage solutions will dictate the final decisions. This paper analyzes the available possibilities to handle these data, starting from languages, protocols, and database management systems based on consideration on updated approaches in these topics. The aim is to develop a solution capable of holding a large amount of both unstructured and structured data that can be published on a web platform accessible for mobile devices.

***Keywords***

*Web-oriented Data Formats; XML; JSON; RDFa; Metadata; Microformats; Microdat; HTML5; DBMS; NoSQL; Internet of Things; Web of Things*

## Introduction

Future Internet vision *(F. Bacelli, et al., 2009)* and its core concepts such as the Internet of Things (IOT) (*D. Uckleman, et. Al., 2011*), the Internet of Services (*C. Schrouth, 2007*) and mobile Internet (*M. Grayson, et. Al., 2011*) form the basis of the future development of the hardware and software infrastructure that will handle different kinds of information and data available primarily for a mobile access. This vision starts from a global network of devices, generally known as objects or things, which link, communicate, and interact with each other as well as other network nodes. Each single object could being a small building block of the Internet, produces a kind of data, specific to the object itself (i.e., information transmitted by sensor nodes (*R. Roman, et. al., 2009*)), but related or linked to other content type (geographical information, social network content, etc.). The aggregated data produced need to be transmitted over network links to be stored, processed, and published on the web. One of the main factors that influences this process, includes how to store data that could have different formats and are often aggregated taking into account the emerging storage and database technologies. Moreover, the same data need to be largely accessible by mobile devices by means of a web publishing methods, which may be in various forms: from web applications, to web services, and web mashups (*S. Pastore, 2010*). Our research focuses on this specific framework by experiment with technologies involved in Future Internet and its concepts to increase the promotion and dissemination of scientific culture. Our research field is the use of Information and Communication Technologies (ICT) for culture in the specific field of Astrophysics and its related sciences (*S. Pastore, 2012*).

This means both the dissemination of scientific activities, projects, and knowledge and the promotion of activities devoted to Astrophysics outreach and education. In this sector, the use of ICT is essential to reach different user audiences consisting of scientists, students, and the public. The aim is to enhance the awareness of future generations of the importance of science and to demonstrate the importance of scientific and technological research in citizens' lives.

The initial idea, summarized in Fig. 1, is to develop and implement a sort of "box", referred to as capture box, consisting of an assembly of hardware and

software modules that collect different kinds of information (i.e., specific to the object to which this box is connected such as astronomical buildings or instrumentations). The box should act as a data collector and then enriched and processed to provide enhanced information on a specific request of the user. The main point is that such information should be accessible mainly for mobile devices usually by means of a wireless communication link.

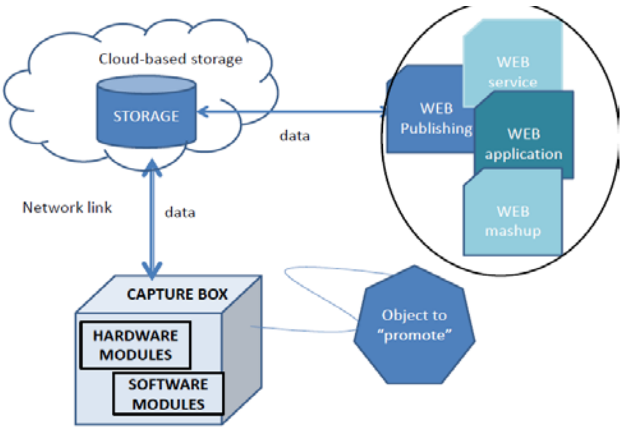


FIG. 1 OUR INTIAL IDEA: THE CAPTURE BOX AND TOPICS RELATED TO STORAGE AND PUBLISH OF INFORMATION

In this paper, focusing on the data, two main issues pertaining to aspects of our implementation are taken into account: how to format and structure the data collected and how to store the data. The first issue requires making specific choices about the format, which should be interoperable and capable to structure different kinds of information. The enrichment of the starting data with metadata before data storage, is analysed. Based on the format used, choices need to be made with regard to the data model. In this category, different aspects of database technologies (*E. Navte, 2010*), and the opportunity to use local or cloud-based solutions (*B. Furth, et. al., 2010*) have been analyzed. It is considered that the data will have to be accessible via the web, taking into account different technologies and software used for web publishing, and that data will be shared, reused, and open.

## Background Research in the Context of Mobile World, Internet of Things, and Web of Things

Technology changes in the Internet and its applications environment are currently influencing the mobile world and moving toward the realization of an interconnected network of objects, which (the things) are usually equipped with a form of wireless Internet-connectivity. Their connections will form a network infrastructure that, when analysed from the network layer it is known as the Internet of Things (IOT), while when analysed from the application layer, it is known as the Web of Things (WOT) (*D. Guinard, et. al., 2009*). Considering the object at the network layer (IOT), it must be accessed directly (by an IP address) and its transmission links use a kind of wireless technology (*G. Gogging, 2008*). Wireless technologies differ on data rate and range (Fig. 2) including short-range transmission (i.e., Near Field Communication (NFC)) and medium range (i.e., Wi-Fi) connectivity. These links are used as a bridge to an Internet link trough wide range technologies (i.e. 3G). Each object can transmit data using NFC or Wi-Fi links.

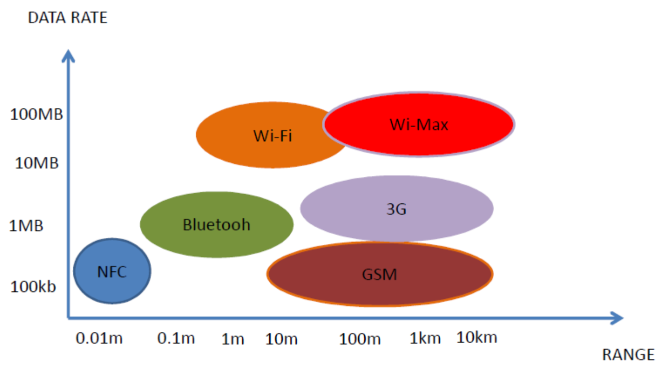


FIG. 2 WIRELESS TECHNOLOGIES CATEGORIZED BY RANGE AND DATA RATE

The "things" include the various categories of objects from mobile devices to specific devices such as sensor nodes. Mobile Internet devices equipped with enhanced capabilities (e.g., Apple iOs-based, Android-based, embedded operating systems or light web servers) provide "an improved object" which contributes to the global infrastructure in terms of both collection and access of the data. According to various statistics about mobile trends (*Mobithinking.com, 2012*), by 2014 mobile devices is more preferential for the connection of Internet than desktop computers. These behaviours are related to the diffusion of newer categories of mobile devices (i.e., smartphone and tablet devices) that are Internet-enabled and sensors-enabled.

From the application layer point of view (WOT), each object, as a building block of the web platform, should be addressed as a web resource using a Uniform Resource Identificator (URI) or a IRI (internationalized Resource identifier) (*W3C website, 2011*). The software

implementation of the object as a web resource is through a web server installed on the object itself. For small devices, the web servers should be lightweight, and an operating system would not be necessary. In this way, the object could export its resources (environmental data, geolocal information) by Application Programming Interface (APIs) utilizing the REST architectural style (*S. Allamaraju, 2010*). This architectural style is based on web resources identified by URI/IRI, available through an uniform interface with well-defined interaction semantics (i.e., HTTP protocol (*D. Booth, et. al., 2004*)) and represented in a self-description representation format. Fig. 3 shows the WOT, with other web trends and technologies that include the social web, the programmable web, the semantic web, and real-time web. These different shades of the web world are all based on the data as the major component of the systems, used in specific applications devoted to people, things and servers.

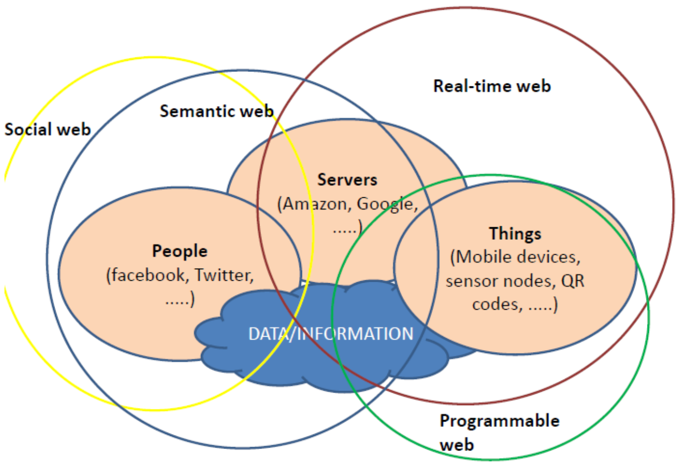


FIG. 3 WEB TRENDS AND WEB TECHNOLOGIES INTERACTING IN THE IOT AND WOT VISION

The information is processed by applications built on top of things and other web services available on the web. Enabling the data aggregation is to develop web mashups that are web application capable to combine data from different sources and present it in several ways. There are several categories of web mashups (enterprise, consumer or data) depending on the specific application domain. The starting point is to aggregate resources from the object (i.e., environmental data) with those provided by external web services (i.e., social content). The information obtained may be enriched with other metadata and processed to obtain several presentations on the web. This process necessarily requires the adoption of standards languages and protocols for the handling and structure data in order to ensure information accessibility and interoperability regardless of the type of device that produces them or that is used to access them and the development platform used for applications.

These standards help structuring the data extracted from the objects and processing the data to be published in a web-oriented way. The idea is that the data should be shared, open, and reusable because web publishing could follow a different framework, which involves different actors: such as search engines, web applications, and web services. In this regard, our focus is on information repositories that impose strict requirements on the data format and on open-source software platforms supporting such data that allow for reusing and sharing.

### *The Project Framework in Information and Communication Technology (ICT) for Scientific Culture*

Our institute as a public scientific research organization has the goal to promote education and outreach science, together with project dissemination in schools and society. Data are intrinsically important in the astrophysical sciences where you are working with a huge amount of data. The astronomical community through specific initiatives formulates standards and recommendations for astronomical data to be used both in the scientific and educational environment (*M. Louys, et. al., 2011*). An example is the variety of non-scientific public image resources coded in formats different from scientific datasets. A project (*R. Hurt, et. al., 2010*) currently define metadata standards to tag these astronomical images (also known as "pretty pictures") used for educational and public outreach. Our research uses astrophysics science as a test environment for the development of our prototype (the capture box) with the aim to prove information to various types of users categorized on various information connected with such science. Our aim is to implement a customized device called a capture box, which is designed as a set of hardware and software modules giving network functionalities, sensor nodes and web resources capabilities. It is assumed that the use of this device is mainly through mobile devices and that both hardware setup and software development must take account of this environment. The capture box is installed (Fig. 4) in the proximity of a specific object (i.e., the target object)

that we want to promote or of which we have more information (i.e., in the early stages of testing the capture box will be installed in the proximity of an ancient or a new astronomical instrument). The box will act as a collector of different data obtained for aggregation that will be transmitted over a wireless network link to be stored in a format suitable to a subsequent processing by different software or applications. Data associated with the object can be applied to different standards, but it is necessary to be linked as they provide a general description on our target object.

The idea is to develop specific applications on top of the capture box as web services performing different tasks (e.g., visualizing data about the status of the environment, presenting general information about the object). These services could be incorporated into other public available web services via a data mashup approach. In this way, customized information is available for users based on individual preferences or interests.

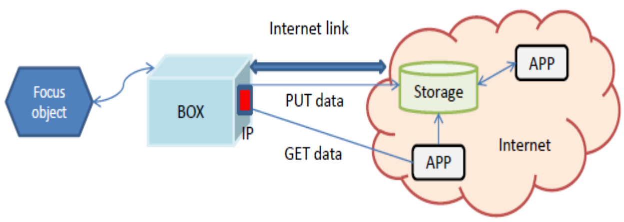


FIG. 4 CONNECTIONS IN THE "CAPTURE BOX"

The information related to the focus object could be in a different format (structured or unstructured data) and should be capable of being archived and stored in order to be searched and found. The idea is to use a software system, mostly a database management system (DBMS) (*E. Navte, 2010*), to collect these data. Factors that need to be considered in relation to the system include the choice of data model, the need to function in a mobile environment, and the need to work on a web platform. Our aim is to exploit modern and emerging techniques in software and technologies to collect, manage and distribute the information. In this paper, we focus on aspects of data structure and storage management solutions to facilitate publishing the data on the web.

## Structuring the Data

The data may be in a structured or unstructured format. Structured data are for example those derived from the sensors (i.e., numerical data), while those defined as unstructured information (*J. Mckendrick, 2011*) are derived from digital formats (i.e., audio, video, graphics, social media messages). A method capable of collecting and managing all these kinds of data is in development, because these different contexts are associated with a single object and thus enriching information. Moreover, the chosen format should be network-aware, because all the information will be transmitted via a network link and then published in some web framework. The use of an interoperable data allows the structuring both data and metadata information that can be associated with these data. The possibility of using non-proprietary formats ensures reusability, availability and readability of data.

The target object is thus described according to different points of view of the users via specific applications. The technologies used for the design should focus on interoperability, ease of processing for better web publishing. These include markup languages such as the eXtensible Markup Language (XML) (*B. Evjen, et. al., 2007*), the Resource Description Framework (RDF) (*R. Roebuck, 2011*), the data-interchange format JavaScript object notation (JSON) (*JSON website, 2012*) or other representation languages that can be compared or derived from the previous formats as the Atom syndication format (*IETF Atom, 2012*).

The main constraints in the choice are related to applications context. In a mobile environment, wireless transmission languages could affect or limit solutions and thus applications are developed as web services or mobile applications. In addition, as the idea is to consider rich data that can be described with different features, the language used should also support metadata information.

XML is a text format, self-describing markup language developed to format and structure data, making the format of choice for interchangeable data serialization. It provides two advantages as a data representation language: text-based, and position-independent. The XML data model is flexible and allows describing data in different ways by defining rules and grammar in a schema. As Fig. 5 shows, XML framework, including a set of technologies able to structure, define, and process data.

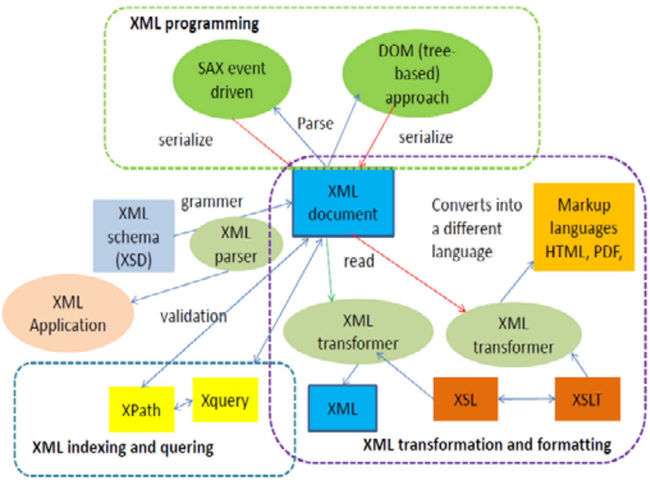


FIG. 5 MAP OF THE XML TECHNOLOGIES FAMILY

At the low-level layer, XML provides a way to format data, in general, by defining a structure made of elements and attributes. The use of specific vocabularies (i.e., namespaces) and the definition of a grammar or schema (XML Schema definition or XSD), allow structuring any kind of information stored in an XML document that is valid according to the scheme adopted. In addition to languages for styling (i.e., transformation and formatting objects with the XML Stylesheet Language (XSL), for querying (i.e., XQuery), for locating information (XPath), and for processing XML documents (parsing methods), are allowed work with homogenous data using technologies that share the same basic concepts.

The parsing of XML documents using a tree-based (i.e., the Document Object Model or DOM) or an event-based (Serial Access XML or SAX) model allows different programming languages to interact with the data.

RDF is data interchange format which is a real data model based on the graph model. It has different syntax (i.e., Turtle, Terse RDF Triple Language, N3 – Notation 3, etc.) of which XML is the one (called RDF/XML). RDF has also several grammars including a schema called ontology represented in RDFS – RDF Schema or Ontology Web Language (OWL) (*R. Roebuck, 2011*), that adds semantics to the data and allows new information to be inferred from the current data. The last examined structuring language, JSON, is considered the competitor language of XML because it has less verbosity to describe the information and hence it is lighter. Also by using Javascript (*D. Flanagan, 2011*) syntax to describe objects, it adheres to the models of data types of programming languages. JSON files are then easily be processed by the web programming languages and then used as the data format for web applications and services. The use of JSON is also prevalent in the mobile world for its lightweight format feature. A comparison between these three formats of data description is provided in Table 1 where the four main features (data model, syntax, security support and processing requirements) are taken into consideration.

TABLE 1 COMPARISON BETWEEN XML, RDF AND JSON FEATURES

| | Data Structuring Languages | | |
|---|---|---|---|
| | **XML** | **RDF** | **JSON** |
| **Data Model** | Tree | Graph | Data Object Notation |
| **Syntax** | Elements/ Attributes | Triples (subject/predicat e/object) | Key/value |
| **Security Support** | Yes | Yes | No |
| **Processing requirements** | High | Medium | Low |

Fig. 6 shows the use of the three languages to describe a specific example object (i.e., a terrestrial globe dated 1839).

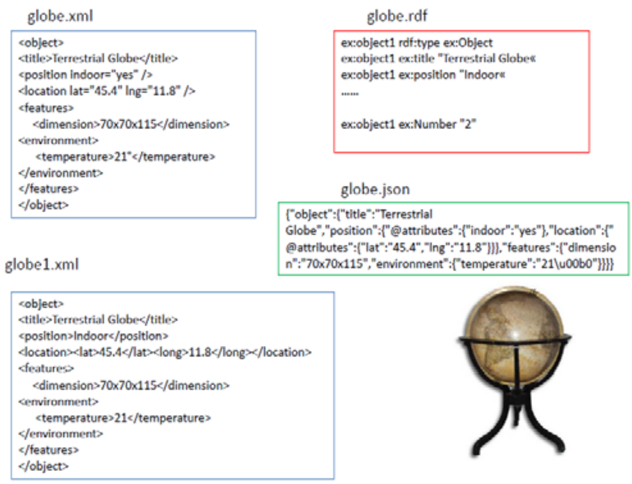


FIG 6 REPRESENTATION OF THE "GLOBE" OBJECT ACCORDING THE DIFFERENT LANGUAGES

The information related to a description of the object, its geographical location and some environmental parameters (i.e., the room temperature), is described as XML elements/attributes, RDF triples or key-value pairs that describe a JSON object.

RDF data has the advantage of simplicity of adding

new information which is solved by adding a specific triple; in XML, it is necessary to change the schema adopted.

Furthermore, while an XML uses unique identifiers for the document i.e., they cannot serve as global identifiers, RDF uses URIs that give a global unique identifiers to retrieve resources. RDF and URI are the technologies that underlie linked data (Bizer, C. et. al., 2009) representing a method of collecting and publishing structured data on the web related to each other by means of links.

The use of semantics facilitates machines to understand the meanings of web information. The use of machine readable meta data extends the network of hyperlinked human-readable web pages. RDF does not have the flexibility of XML, but conveys statements about things. Both RDF and XML fulfil the task of defining the relationship of data, but RDF is supposed to be able to define what the relationships between these data. Finally JSON describes the information in a form that can be easily processed and thus is less complex.

Focusing on these three main languages, there are advantages and disadvantages on their use. XML, for example is easier to work with because it includes many supporting technologies. Its main problem is, in some way, the verbosity of the language, which impacts data-interchange because it carries a lot of baggage and does not match the data model of most programming languages. JSON uses a simplified syntax, more lightweight and seems to be predominant in the mobile domains. However, it lacks flexibility in terms of security options. RDF allowing a resource to be described is used in the semantic web, but usually public web services are based on XML or JSON representation. Fortunately, there are tools available to convert between these data. The converters (i.e., XML to RDF (*D. Wood, 2011*) or JSON to XML (*D. Lee, 2011*), shown in Fig 7 could make it possible to work with these standards.

By considering these aspects and the presence of different kinds of data that could enrich the single object, it seems reasonable to have the information available in these formats. The transformation would be relatively easy. Applications that process the data (i.e., web services, applications or mashups) are developed by using the best format according to the type of architecture chosen.

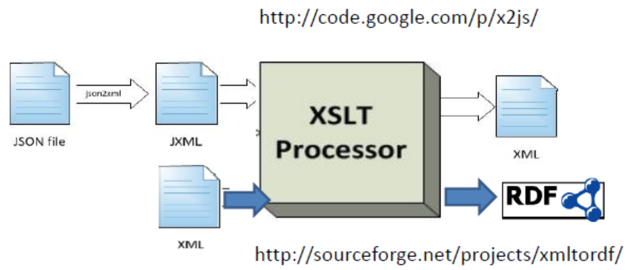


FIG. 7 CONVERSIONS BETWEEN LANGUAGES BASED ON XML TECHNOLOGIES

Focusing on methodologies, web resources structured in one of these languages are accessible as web services. In mobile context, the architectural style chosen to develop services is resource oriented that is the use of Representational State Transfer (RESTful) API (*G. Reese, 2012*) to interface with resources. In the REST paradigm, resources represented in specific network languages such as JSON for processing goals, or XML for integration with business systems; which are named with an URI; and linked together, offer an uniform interface by means of the HTTP protocol and its methods. Reception and transmission of messages between the object and the web, pass through the HTTP protocol and its verbs (i.e., GET to retrieve the representation of the resource, PUT to update a resource) or its headers (i.e., metadata that defines what kind of data is sent and received or the connection status using status codes). A REST-based approach opposite to the SOAP (Simple Object Application Protocol)-based web services architecture (*G. Alonso, 2007*), has given rise to the resource oriented architectures (ROA) that is a direct application in the mobile and web of things contexts (*B. Sletten, 2009*). The complexity of the SOAP-based web services (often called big services) approach which is based on a protocols stack that implements the several layers of messaging between a service provider and a consumer, is lightened in the development of RESTful web services that use the transport protocol of the web (HTTP) to make calls between machines. Software implementations follow various programming languages even it seems to prefer web programming languages (i.e., PHP, Python, Ruby). The support towards the different kind of data formats does not have impact on the decision on the implementation of software to choose for the development of the applications on top of the resources' object. Instead the metadata format of the information that can be attached to the general data may be a constraint whereas the processing of this information. The data will usually be published in a

web format that will result in a web page structured with the HTML language. Embedded metadata in information when it appears as a web page, and then processed and presented by a web browser, helps to enrich content and link with other applications or services. Metadata are usually added within a web document using elements (tags) and attributes of the HTML language that are then processed by the web browser engines. The methodology associated with the use of metadata is more frequently used in the development of web applications when it allows greater integration with other applications or services even in the mobile world. Between the formats used to define metadata information, RDF attributes (RDFa) (*K. Roebuck, 2011*), microformats (*Microformats website, 2012*) and HTML5 microdata (*WHATWG website, 2012*) are the most used to enrich the information related to individuals, organizations, product. They are also catching on metadata associated with the social world of which the Facebook's Open Graph Protocol (OGP) (*OGP website, 2012*) is an example, which then allow connecting with these networks. Metadata are usually defined in specific vocabularies: each of which has its own vocabulary, but it is possible to use other grammars such as those defined by schema.org (*schema.org website, 2012*)) which thus allowing a better description of the content.

### *Embedded Metadata in Data Structure Languages*

Embedded metadata specifications allows content to be annotated with specific machine-readable labels and processed by software. Web pages enriched with such content could be easily machine readable, as well as human readable. The specifications have different methods of application within an HTML page, but still use HTML elements (i.e., div or meta) and HTML attributes (i.e., class) that are also used in connection with the language of presentation of data or style sheet language (CSS) (*P. Gasston, 2011*). RDFa adds a set of attribute-level extensions to web documents (i.e., HTML, XHTML or XML-based) and allows including different schemas or vocabularies. It provides a set of attributes (i.e., *property* to specify the properties of the element content, *about* to define the URI of the resource, *rev* and *rel* to specify a relationship with another resource) used to describe HTML tags (i.e., *span*, *div*, *p*). Microformats are particular specifications designed as a set of open data formats that describe some content such as people, organizations, events, locations, and products; each of which (i.e., *hCard, hProduct, hPerson*) has a set of properties used in the description of an HTML element (i.e., *div*) through the specification of its attributes (i.e., *class*) as the value associated to a specific attribute (i.e., *class="fn"*). Various microformats are modular on one web page. Fig. 8 shows an example of enrichment of information related to our target object (i.e., the terrestrial globe) with microformats. *hProduct* describes the object, *hRewiew* describes the impressions about the object by visitors; and *geo* describes the location of the object.

```
<div class="hproduct">
<h1 class="fn">Terrestrial Globe</h1><img class="photo" src="glogter.jpg" alt='Terrestrial Globe' />

 Product description:  <span class="description"> Date:1839; Joseph Juttern, Vienna. Wood, paper.
Diameter: 63cm. Dimensions: 70x70x115cm </span>
<span class="geo">Location
<span class="latitude"><span class="value-title" title="45.402750" />
<span class="longitude"><span class="value-title" title="11.869378" /></span>
</div>

<div class="hreview">
<span class="item">
<span class="fn">Terrestrial Globe</span></span>
Reviewed by <span class="reviewer" >Ellie</span>on

<span class="dtreviewed">April 1<span class="value-title" title="2011-04-01"/></span>
<span class="summary">Good</span><span class="description">No comment</span>
Rating: <span class="rating">3.5/5</span>
 </div>
```

FIG 8. REPRESENTING THE TARGET OBJECT WITH MICROFORMATS EMBEDDED METADATA.

Microformats are a method that has been successful for their simplicity and the absence of specific schemas. The downside is that with this syntax, semantic information is combined with the styling. Thus, it does not complement good web design. With HTML5, the WHATWG group implementing the specifications related to such language, decides to create its own version of an interconnected information organization. HTML5 Microdata is a new feature of the HTML5 specification to embed semantic information within web pages.

Microdata defines a vocabulary (which is located at http://data-vocabulary.org) that includes a collection of descriptions of persons, events, organizations, products, which are defined as item. Each item (i.e., *Product*) is defined within HTML tags using the *itemscope* element that is normally associated to the HTML *div* tag. *Itemscope* is described with the *itemtype* attribute that specifies the URI connection to the vocabulary used (*itemtype="http://data-vocabulary.org/Product"*). Global attributes of the item (i.e., *itemscope, itemtype, itemprop*) allow the description on an item by its creation (*itemscope*), the schema definition adoption (*itemptype*) and the values of its properties (*itemprop*). Microdata make the use of different vocabularies such as those provided by schema.org. The schema.org vocabulary provides a set of common types of items from the *Thing* item type,

and has also inherited, a number of properties that describe the item.

Shown in Fig. 9, our target object is described using microdata specification and the schema.org vocabulary.

**Microdata Schema.org →**
itemtype=http://schema.org/Product

*Original HTML description*

```
Terrestrial Globe <img src="globter.jpg" alt=Terrestrial Globe' />
Rated 3.5/5 based on 1 visitors reviews
Location: Meridian room
Product description:
Date:1839; Joseph Juttern, Vienna. Wood, paper. Diameter: 63cm. Dimensions: 70x70x115cm
Visitors reviews:
Good - by Ellie, April 1, 2011. No Comment
```

*With microdata*

```
<div itemscope itemtype="http://schema.org/Product">
<span itemprop="name">Terrestrial Globe</span><img src="glogter.jpg" alt='Terrestrial Globe' />
<div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AggregateRating">
Rated <span itemprop="ratingValue">3.5</span>/5 based on <span
itemprop="reviewCount">1</span> visitor reviews</div>
 <link itemprop="availability" href="http://schema.org/InStock" />Meridian Room</div>
  Product description:  <span itemprop="description"> Date:1839; Joseph Juttern, Vienna. Wood,
paper. Diameter: 63cm. Dimensions: 70x70x115cm </span>
 Visitors reviews:  <div itemprop="review" itemscope itemtype="http://schema.org/Review">
   <span itemprop="name">Good</span> - by <span itemprop="author">Ellie</span>,
   <meta itemprop="datePublished" content="2011-04-01">April 1, 2011
   <div itemprop="reviewRating" itemscope itemtype="http://schema.org/Rating">
    <meta itemprop="worstRating" content = "1"><span itemprop="ratingValue">1</span>/<span
itemprop="bestRating">5</span>stars   </div>
   <span itemprop="description">No comment</span>
  </div> </div>
```

FIG 9. REPRESENTING THE TARGET OBJECT BY THE INCLUSION OF METADATA SCHEMA.ORG INFORMATION.

The *itemscope* is added to the *div* tag to enclose the information about the item that is defined as a product. The *itemtype* attribute associated to the *itemscope* specifies the use of the schema.org vocabulary. The properties associated to the item, entering to specify an HTML element like *span*, are labelled using the itemprop attribute in the usual form *attribute="value"* to describe for example the name of the product/object (*itemprop="name"*). Other items used located in the same vocabulary are for example *Rewiew* or *Rating* that allow visitors, always using the *itemprop* attribute, respectively to make a judgment on the object through a sentence (*itemprop="name"*) and leave their name (*itemprop="author"*) or a vote (*itemprop="RatingValue"*).

Metadata syntax can usually be added inside a web page by means of different attributes labelling the information for machines. It needs a proper namespace (a specific schema), the inclusion of such schema together with the declaration of the information type. The downside of newer approaches (i.e., HTML5 microdata) could be the novelty and the lack of full support in all browsers.

Finally, among other approaches to the description of the information with metadata, the Facebook OGP links content to social networks information. Similar to other specifications, this defines its vocabulary of elements and attributes (http://ogp.me/ns#) usable within a web page. The basic protocol idea is to integrate web pages into the social graph via structured data. The protocol, in contrast to previous specifications, adds information to the HTML *meta* tag (that is a tag of the header part) by specifying a range of properties used as attribute of such tag. The properties always written in the form *attribute="value"* use the *og* prefix to refer to the namespace. The protocol defines a set of required properties (i.e., *og:title, og:type, og:image, og:url*), to define the context of the object along with other optional properties that specify other information related to that object (i.e., location). OGP supports different object types to describe the content through the og:type property including activities, business, people, organizations. Fig. 10 shows the description of the object enriched with information related to the social world using the OGP protocol.

property="og:type" content="product" →
**Facebook Open Graph product**

```
<html xmnls="og:http://opengraphprotocol.org/schema">
<head>
<meta property="og:type" content="product" />
<meta property="og:title" content="Terrestrial Globe" />
<meta property="og:image" content="globter.jpg" />
<meta property="og:description" content="Date:1839; Joseph Juttern, Vienna. Wood,
paper. Diameter: 63cm. Dimensions: 70x70x115cm" />


</head>
<body>
<h1>Terrestrial Globe</h1>
....
</body>
```

FIG 10. REPRESENTING THE TARGET OBJECT BY THE INCLUSION OF SEMANTIC INFORMATION FOLLOWING THE OPEN GRAPH PROTOCOL

The definition of the vocabulary used is made by html tag. After the object is described within the *meta* tag as *og:type* property (*property="og:type"*) with the value defined using the *content* attribute (*content="Product"*). Various *meta* tags are used to describe other properties of the object (i.e., description, image) seen as a product through the attribute *property* that specifies the property and *content* that specifies the value (*property="og:description" content="Date:1836"*).

The decision related to the most appropriate formats depends on several factors and, in some cases, on different opinions. For some, the question is which format is better from an elegance-of-coding perspective. For others, the issue is the simplicity of the solution or the most cost-effective solution to apply and least likely to cause problems. However, as our focus is on global use, the most appropriate choice is probably the microdata solution, together with different schema and the inclusion of social network

based metadata (i.e., OGP).

### *Our Decision About Web-oriented Formats and Their Management*

From a research perspective, the best solution is to adopt different languages and schemas and to test their deployment and to determine what protocols are supported. Whatever decision on data formats to the mobile world (JSON vs. XML) is supported by the fact that data conversion is possible. Thus we can start by means of referring to web resources and using URI/URL to identify any item on the web in a universal way. The basic concept is that the web is a graph of information with nodes (web pages) and edges (link) that connect all the information together. Thus, each resource is described by a structured language, as well as by the information from the embedded metadata i.e., the invisible information in the HTML document. This information will be machine readable but will not been seen in the browser. In this way, the information can be exported and processed by web services. We can define the following steps as a method to work with data:

- Data/content structured in XML-like formats and use of conversion tools to have different representation of the same object (i.e., XML, RDF and JSON structure);
- Adding metadata for semantic information according to several formats (i.e., microformats vs. microdata with different schemas and OGP support);
- Storing data on-line in order to retrieve, search and publish information in web format.

The decision algorithm flowchart is shown in Fig. 11.

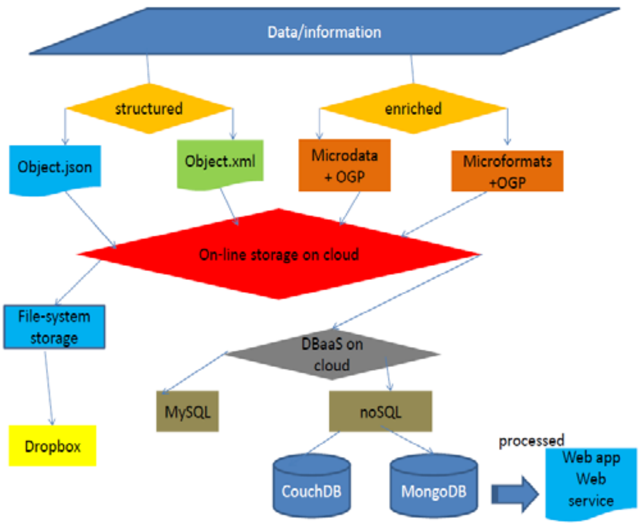


FIG 11. THE DECISION ALGORITHM FLOWCHART

We start with a collection of information about our target object from different sources (i.e., directly by the capture box or indirectly by the web platform) that may be available in various formats. Some of this content is enriched with metadata information or metadata (i.e., HTML5 microdata or OGP metadata) added to the HTML templates that are used in the publication of information as web pages. The process of collection and aggregation of information on the web, in a customized and mobile way, must be preceded by data storage.

So before developing mobile applications that present the information customized for users, it is necessary to take a decision on the method of data storage to be implemented. Online solutions guarantees availability of the content along with the other features required for a storage solution. Currently solutions are based on the cloud offering opportunities that shall be released from the management of complex systems and allow immediate access to the information stored. The cloud paradigm provides a set of opportunities to use computing facilities in a broad sense (i.e., applications, services, virtual machines, etc.) with scalability features.

While choosing a cloud environment, there is a choice between a storage solution based on file system storage or based on database storage. If we choose the second option, then we will have to evaluate the different models of databases and database technologies used for data management. In particular, we must take account of new technologies related to the NoSQL (*S. Tiwari, 2011*) approaches or the use of non-relational database that are based on the standard SQL language to interact with the data.

## Storage of Web-oriented Data Using the Cloud Paradigm

On-line storage is a requirement in our context. There is a need to store the data transmitted by our "capture box" and the other data and metadata related to our focus object. The box will have limited storage capacity. Thus, all data structuring and processing should be take place using hosts accessible via the Internet connection. The cloud seems an appropriate distributed platform to store and process the data, offering the opportunity to work on-line and providing scaling opportunities when the amount of data grows.

The first decision is what kind of storage should be used: a file system approach or a database

organization approach. The cloud-based solution may be appropriate for both approaches.

If we consider a database solution, which would enable us to take advantage of features of database management systems such as indexing and querying, we confront another decision regarding the type of technology to be used. In database technology, NoSQL database management systems offer many solutions. This type of technology is extensively used in social network frameworks due to its capacity to handle data storage, interconnected data and complex data structure. NoSQL systems are designed to solve new requests arisen from big data and agile programming patterns [36]. The cloud environment is also suitable for different types of database-management solutions.

### *Type of Storage: File System or DBMS?*

Cloud-based storage solutions include the storage on a filesystem-like and a set-up using a database management system.

The unique features of a cloud database, namely the ability to distribute data across wide geographical areas and among different servers in one physical data center, are based on cloud computing technology making possible by virtualization. The storage in the cloud as filesystem storage offers many options. For example, there are different cloud storage services with varying price-per-gigabyte, and these have various features and functionality.

Table 2 shows a list of various cloud services that could be adopted.

TABLE 2 LIST OF CLOUD STORAGE SERVICES

| | Features | |
|---|---|---|
| | **Storage options** | **Access** |
| **Amazon Cloud Drive** | 5 GB/free, additional $10 per year; | Web/android |
| **Box.net** | Different for individuals (free 5/50GB) and business (1000GB 13euros per users per year; | Web, iOS, Android |
| **Dropbox** | 2GB/free, business plans at 100, 200, 500GB starting $9.99/month | Windows, Mac, Ubuntu, web, iOS |
| **Google Drive** | 5GB/free and various prices | Web, android, mobile web |
| **iCloud** | 5GB/free, and then different prices | Mac, IoS, web |
| **Microsoft SkyDrive** | 25GB/free | Mac, web, Win, Mobile web |

Most of them offer a certain amount of storage for free. In the first implementation of our design, we could choose one solution with access devoted to a mobile device.

However, file-system based implementation would not be sufficient in our case because all features related to the adoption of the database structure would be lost. Instead, a solution based on a database structure is preferable to our project. It is necessary to aggregate and query the content, processing the request, and retrieve data in a web-based format. The next decision will be the type of data model to be applied by considering the type as well as the use of the data.

### *DBMS Category: SQL or NoSQL*

Newly emerging data management and database technologies are gradually becoming mainstream alternatives to traditional relational databases. Database systems are gaining popularity because the volumes of data requiring storage have increased massively. The historical data model based on relations or object relations is being replaced with other models such as graph-oriented, column-oriented, document-oriented, schema-less, and key-value. The so-called non-relational or NoSQL databases are gaining popularity as alternatives to conventional database management. NoSQL represents a set of technologies called structured storage that do not fit in the relational paradigm. The NoSQL approach has emerged as a result of a need of a better management on large amounts of data storage, the need for interconnection of data, and the complex nature of the data structure, e.g., nested data. Data stored in a NoSQL system can be structured. The main difference is that this kind of database is not based on the SQL language for querying data. However NoSQL databases have different methods to index and search information in the storage. Such database systems guarantee consistency but no other main features of the relational databases that remain the best solution in specific contexts. The key aspect of the NoSQL system is its distributed and fault-tolerant architecture.

The current NoSQL schema fits into four categories shown in Table 3, and there are several softwares that implement these categories.

The features of NoSQL databases are different depending on the category and the software implementation, but still share a common set of key characteristics that are horizontal scalability, support data replication, and based on HTTP/REST protocol.

TABLE 3 NOSQL CATEGORIES AND SOFTWARE IMPLEMENTATIONS

| Category | Features | Software implementation |
|---|---|---|
| **Key-value stores** | An hash table where there is an unique key and a pointer to a particular data | Amazon dynamoDB , MemcacheDB |
| **Column family stores** | To handle large amounts of data distributed over many machines. There are keys that point to multiple columns | Apache Cassandra, Apache Hadoop |
| **Document database** | store documents that are collections of other key-value collections. The semi-structured documents are stored in formats like JSON | Apache CouchDB , MongoDB |
| **Graph database** | are built with nodes, relationships between nodes and its properties | Neo4J |

With regards to different software implementations, Dynamo, created by Amazon.com, is the most prominent key-value NoSQL database. Open-source NoSQL database such as Hadoop's Hbase, Cassandra, CouchDB or MongoDB have replaced relational databases used by social networking sites and cloud platform providers. These databases using programming interfaces that are more web-friendly, having built-in replication capabilities, exhibit high availability, are resistant to crashes and corruption because of the way they are built. Their main features are the ability to synchronize with remote databases (e.g., CouchDB). However, despite the ever-increasing bandwidth of mobile networks, they are inherently unreliable. A loss of connection to data is compensated by a local cache of data, enabling mobile applications and web technologies to continue to operate when the network does down. NoSQL databases are also generally designed from the ground up to require less management data and lower transaction volumes. Thus, the cost per gigabyte or transaction/second can be many times less that the cost for relational database management systems. These features could make them attractive option, even if the choice of the category is related to the kind of data store. For example, we could use a document database model for our data with an XML document view or a JSON file view. Software implementations in this context could be MongoDB or CouchDB.

### *Cloud-based DBaaS Solutions*

Database as a Service (DBaaS) (*A. De Monroy, 2011*) is a database solution in a cloud environment. DBaaS refers to the provision of database utilities from a server in the cloud environment, where databases are provided to users according to a payment model based on the effective utilization of resources. The cloud DBaaS is categorized following the kinds of services offered software, platform, and infrastructure: SaaS, PaaS, and IaaS respectively (*C. Weinhardt, 2009*). The database utilities could be provided via remote access to the database or via a specific a virtual machine providing a database management system. In relation to cloud-based implementation, there are various database solutions in the cloud that implement both the relational and the non relational paradigm. Some of these are listed in Table 4. It can be seen that the same vendor, e.g., Amazon, offers both solutions (with a MySQL implementation and the SimpleDB (*Amazon SimpleDB website, 2012*) implementation).

TABLE 4 SQL AND NOSQL CLOUD DATABASE SERVICES

| SQL Solutions | NoSQL solutions |
|---|---|
| Microsoft SQL Azure (MS SQL server) | Amazon DynamoDB (DynamoDB) |
| Xeround (MySQL) | Amazon Simple DB (use hash storage) |
| Amazon Relational Database Server (MySQL) | CouchBase (CouchDB) |
| Enterprise DB (Postgres) | MongoHB, MongoOd grid (Mongo DB) |
| Heroku (Postgres) | Nuvolabase (GraphDB) |
| ClearDB (MySQL) | Cloudant (CouchDB) |

For our project a cloud solution may provides a good way to store and process our data, because it avoids the costs related to a local system management. Between alternatives models of databases, we focus on a NoSQL solution, considered the use of these technologies in the web world. A possible adoption could be provided by vendors like Amazon with SimpleDB, or Google with the Google App Engine Datastore (*GAE website, 2012*), or probably a MongoDB implementation. Since however proprietary solutions seem to be, in some way, linked to the platform used (e.g., an application in the Google App Engine datastore is hardly portable in another platform than Google App Engine), we decided to choose a solution related to open source implementations as MongoDB or CouchDB. We found various hosting service providing NoSQL cloud-based solutions. Examples of cloud-based providers for the two NoSQL software implementation (e.g., MongoDB and CouchDB) are MongoLab (*MongoLab website, 2012*) or IrishCouch (*IrishCouch website, 2012*). Both these solutions offer a possibility to work with the system without any cost for testing purposes. Our aim is to develop

applications able to interact with data stored in such online database, to demonstrate the efficacy of the solutions. Thereafter it may be able to evaluate the costs and services offered by such platforms depending on the requirement of our system.

## Conclusions and Further Improvments

Structuring data in a web-oriented way allows information to be readily published and reused, especially when the mobile environment is in consideration. In this paper, we analyzed new trends in the evolution of network architecture and underlying technologies that are evolving toward the Internet of things and the Web of things to implement solutions in the framework of culture dissemination. Our project aims to aggregate and collect relevant information pertaining to a target object that we propose to promote. Such information is thus processed in order to be published on the web platform. Information is obtained and used mainly through mobile device and mobile technologies. Moreover data published will be accessed by a mobile browser or application.

The collecting device is an embedded device that we called capture box implementing a “thing” in the IOT perspective. Data obtained from this object and other information linked to it. Our intention is to structure collected data using open and interoperable language that allows for publishing with the different web tools. We considered the various languages: XML, JSON or RDF, and decided to use different representation in each of such languages according to the kind of content collected, since the conversion tools help such task. Because the goal is web publishing, the introduction of semantics using embedded data such as metadata that can be easily included in web languages, could enrich the information, help the discovery, its linking, and its reuse. We propose to use HTML5 microdata enhanced with other vocabularies like schema.org and to use semantics of the social graph with OGP. Aside from the structuring and the processing of the information for the publishing, the next consideration is the storage and management of data. We consider a appropriate cloud-based solution due to the availability of different software implementations. A database solution as a backend tool is a key component of the major applications development frameworks. In our project, it may be possible to exploit emerging technologies in database management and the advent of NoSQL solutions. The data management system we propose could be an attractive approach considering that it works better in modern computing platforms such as clouds. Among the different categories of these database systems, a document-based noSQL system could be our storage system as it meets the main requirements of our data. We decided to test two cloud database hosting solutions freely available to test purposes of MongoLab and Iris Couch. These databases, that are schema-less, work on data organized into collections that are indexed and searchable. Our variety of data is better stored and searched in these collections. The next step will be the software implementation of the technologies, starting from a hardware prototype of the capture box, and the development of an application able to present data related to our target object.

## REFERENCES


Allamaraju, S. “RESTful web services cookbook: solutions for improving scalability and simplicity”. Yahoo Press, 1ed. 2010.

Alonso G., Casati F., Kuno H. , and Machiraju V. “Web services: concepts, architectures and applications (data-centric systems and applications). Springer, 2010.

Amazon DynamoDB. Available http://aws.amazon.com/dynamodb. Accessed November 30, 2012.

Amazon SimpleDB website. Available: http://aws.amazon.com/simpledb/. Accessed November 28, 2012.

Apache Cassandra project Available http://cassandra.apache.org. Accessed November 30, 2012.

Apache CouchDB website. Available http://couchdb.apache.org. Accessed November 30, 2012.

Apache Hadopp website. Available http://hadopp.apache.org. Accessed November, 30 2012.

Bacelli, F., and Crowcroft, J. “Future Internet Technology”. ERCIM Magazine, April 2009, pp.16-18.

Bizer C., Heath T., and Berners-Lee, Tim. “Linked data – The story so far”. International Journal on semantic web and information systems (IJWIS), Vol. 5, No. 3, March 2009, pp.1-22.

Booth, D, Haas H, McCabe F., Newcomer E., Champion M., Ferris C, and Orchard D. “Web Services Architecture”. W3C Working Group Note, February 2004. Available:

http://www.w3.org/TR/ws-arch. Accessed November, 28 2012.

De Monroy, A. "The pros & cons of DBaaS". Available: http://thecloudtimes.com/2011/11/27/the-pros-cons-of-dbaas/. The cloudtimes.com. Accessed November 28, 2012.

Evjen B., Sharkey K., Thangarathinam T., and Michael K. "Beginning XML, 4th edition". Wrox, 2007.

Flanagan, D. "Javascript: The definitive guide". O'Reilly Media. 2011.

Furth, B and Escalante A. "Handbook of cloud computing". Springer Science. LLC 2010.

Gasston, P. "The book of CSS3: a developer's guide to the future of web design". No Starch Press, 2011.

Goggin, G., and Hjorth, L. "Mobile Technologies: from telecommunication to media". Routledge 1ed, 2008 (11)

Grayson, Shatzkamer, and Wierenga. "Building the Mobile Internet". Cisco Press, 2011.

Google App Engine Datastore website. Available: https://developers.google.com/appengine/docs/python/datastore/. Accessed November 27, 2012.

Guinard, D. and Vlad, T. "Towards the Web of Things: Web Mashups for Embedded Devices". In Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in Proceedings of WWW (International World Wide Web Conferences). Madrid, Spain, 2009.

Hurt, R. and Christensen, L.L. "Making Images Smart: Virtual Astronomy Multimedia Project Astronomy Visualization Metadata". American Astronomical Society, 2010.

IETF organization website. "Atom syndication Format". Available: http://www.ietf.org/rfc/rfc4287.txt. Accessed October 10, 2012.

Irish Couch website. Available: http://www.irishcouch.com. Accessed November 30, 2012.

Javascript Object Notation (JSON) website. Available: http://www.json.org/. Accessed October 10, 2012.

Larrucea X., and Bozheva, T. "Towards an agile process pattern modeling framework". In Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering (SE'07), 2009, pp.61-65.

Lee, David A. "JXON: an An Architecture for Schema and Annotation Driven JSON/XML Bidirectional Transformations." Presented at Balisage: The Markup Conference 2011, Montréal, Canada, August 2 - 5, 2011. In Proceedings of Balisage: The Markup Conference 2011. Balisage Series on Markup Technologies, vol. 7 (2011). doi:10.4242/BalisageVol7.Lee0

Louys M., Richards A., Bonnarel F., Micol A., Chilingarian I and McDoewell J. "IVOA recommendations: Data Model for Astronomical Dataset characterization". 2011 Accessed September 14, 2012 Available: http://www.ivoa.net/Documents/REC/DM/CharacterisationDM-20080325.pdf.

Mckendrick, J. "Unstructured data: the elephant in the big data room". Zdnet blog. Available: http://www.zdnet.com/blog/service-oriented/unstructured-data-the-elephant-in-the-big-data-room/7116. Accessed on November, 39 2012.

Memcache DB: a distributed key-value storage system website. Available http://memcacheddb.org. Accessed November 30, 2012.

Microformats website. Available: http://microformats.org. Accessed November 30, 2012.

Mobithinking.com website. "Global mobile statistics 2012". Available: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats. Accessed October 10, 2012.

MongoDB website. Available: http://www.mongodb.org. Accessed November 30, 2012.

MongoLab website: MongoDB hosting. Available: https://mongolab.com. Accessed November 27, 2012.

Navte, E. "Fundamentals of database systems". Pearson education, 2010.

Neo4j website: World's leading graph database. Available: http://neo4j.org. Accessed November 30, 2012.

Open Graph Protocol website. Available: http://ogp.me. Accessed November 30, 2012.

Pastore, Serena. "Is the Mashup technology mature for its application in an institutional website?". Procedings of the 4th International Conference on Mobile Ubiquitous Computing Systems, Services and Technologies (UBICOMM 2010), 25 October 2010. IARIA Press, pp. 351-356.

Pastore, Serena. "E-business and Research Institutes: when technologies, platforms and methods converge in providing web applications and services to meet users' needs". E-Business- Application and Global Acceptance, Edited by Princely Ifinedo, Feb. 2012.

Reese and Reilly C. "The REST API design Handbook". Kindle edition, 2012.

Roebuck K. "Web 3.0 – The semantic web: high-impact strategies – what you need to know: definitions, adoptions, impact, benefits, maturity, vendors". Tebbo. 2011.

Roman, and Lopez. "Integrating Wireless Sensor Networks and the Internet: a Security Analysis". Internet Research, Vol. 19, no. 2, pp. 246-259, 2009.

Schema.org website. Available: http://schema.org. Accessed November 30, 2012.

Schroth, C. "Web 2.0 and SOA: converging Concepts enabling the Internet of services". IT Professional Journal, IEEE Computing Society, Volume 9, Issue 3, May-June 2007, pp. 36-41.

B. Sletten, B. "Resource Oriented Architecture: The Rest of REST". InfoQ.com. 2009. Available: http://www.infoq.com/articles/roa-rest-of-rest. Accessed 30 November 30, 2012.

Tiwari, S. "Professional NoSQL". Dinam, Wrox, 2011.

Uckleman, Harrison, and Michahelles. "An Architectural Approach Towards the Future Internet of Things". Architecting the Internet of Things Book, Springer-Verlag Berlin Heidelberg, 2011. p. 1-24.

W3C website. "Note on URIs, URLs and URNs: Clarifications and Recommendations 1.0". Available: http://www.e3.org/TR/uri-clarification, 2011. Accessed November 30, 2012.

W3C website. "RDF and JSON". Available: http://www.w3.org/blog/SW/2011/09/13/the-state-of-rdf-and-json/. Accessed September 12, 2012. (26)

Weinhardt C., Anandasivam W.A., Blau B., Borissov N., Meinl T., and Michalk W.W."Cloud Computing – a classification, business models and research directions". Business and Information System Engineering, Vol. 1, Issue 5, October 2009, pp. 391-399.

WHATWG site, Microdata specification. Available: http://dev.w3.org/html5/md/ (2002). Accessed November 30, 2012.

Wood, David. "The State of RDF and JSON". W3C blog. Available: http://www.w3.org/blog/SW/2011/09/13/the-state-of-rdf-and-json. 2011. Accessed November 30, 2012.

**Serena Pastore** is an electronic engineer specialized in ICT working as researcher at the Italial National Institute for Astrophysics. She has over 10 years' experience in ICT projects applied to technology for astrophysics research and for education, dissemination and communication of science. She is an expert in distributed network paradigms (i.e. grid and cloud since worked in the European Grid Infrastructure projects from 2003), Internet technologies and standards for the web, wireless standards and e-content for mobile devices. She is also a professor of computer science subjects (i.e. xml languages, operating systems and networks, web design, web programming) in some Universities (i.e. Ca' Foscari University of Venice, University of Padova, University of Trento). She has published several scientific papers on international journals and conferences on the above mentioned topics. Additionally to that she also covers areas as EU project planning and infrastructure project management..